Capstone Report

Machine Learning Engineer Nanodegree

# Definition

## Project Overview

A stock market , as defined by Wikipedia, is 'the aggregation of buyers and sellers of stocks, which represent ownership claims on businesses.' In effect they are markets where individuals or corporations can buy and sell shares of a business.

The prices of the stocks are determined by the market, or in other words the price varies constantly depending on what people are willing to buy or sell that stock for. If you are able to predict how the price of stocks will change over time then it would be possible to buy when the price is low and sell when the price is high making large profits in the process.

There are many companies and individuals that try to predict how the market is going to change to make money. Most of these perform analysis on a variety of data to get as accurate predictions as possible to minimise risk and maximise profits. There are two main schools of thought on the best data to use to try and predict the stock market. These are fundamental analysis and technical analysis.

Fundamental analysis concentrates on determining the companies value based on information such as its income statement, balance sheet and cash flow statements. When a company is trading below its intrinsic value it is considered a good investment opportunity.

Technical analysis is unconcerned with the companies value because it is believed that the stock price already contains all the relevant information. It therefore uses only historical stock market data and various statistical techniques to try and determine how a stock's price is going to change in the future.

Technical analysis tends to be focused more on short term trading (over weeks, days or less) whereas fundamental analysis is more geared towards long term trading (months, years or longer).

This project is going to look at using technical indicators to try and predict whether a various stock's prices will go up or down over the next day (i.e. the direction of the daily return) and ultimately use this information to inform a trading strategy. This will then be compared against a model that also uses external information from twitter to help inform its decision. The aim is to determine whether the external information can improve a predictive model.

## Problem Statement

The problem that this project is trying to solve is whether information from twitter can be used to improve a predictive stock trading algorithm.

There are three main steps that are required to answer this question, they are:

1. Build and train a model on technical indicators calculated from historic stock market data to predict the direction of daily returns of stocks.
2. Build and train a model on technical indicators calculated from historic stock market data and twitter data to predict the direction of daily returns of stocks.
3. Compare the accuracy of the two models and compare how they perform over a period of time with simulated trades.

## Metrics

Two different metrics will be used during this project. The first will be used to determine how good the different models are during training and tuning of the models. The second will be used to compare the effectiveness of the models when used for trading.

As the models predict the direction that the stock prices is going to go (up or down) the algorithms used will be classifiers and hence the metric used to score different models will be accuracy. This measures how often the model predicts the correct answer and is defined mathematically as follows:

Where:

|  |  |
| --- | --- |
| *n* | Is the number of samples |
| *1()* | Is the indicator function |
| *ypred* | Is the predicted value |
| *yactual* | Is the actual value |

After the models have been trained they will be used to inform a trading strategy. The Return On Investment (ROI) of the trading strategies, over a fixed period of time with a fixed initial capital, will be used as a metric to compare the different models. The ROI is defined as follows:

On top of comparing the models created they will also be compared against the standard benchmark of a 'buy and hold' strategy on the S&P 500. The S&P 500 is an American stock market index based on the market capitalisation of 500 large companies having common stock listed on the NYSE or NASDAQ . It is commonly considered the best representation of the US stock market as a whole and hence represents how the entire stock market varies on average over time. Comparing the models against this will determine whether they would be able to 'beat the market'.

# Analysis

## Data Exploration

### Stock Market Data

Daily historical stock data can be easily downloaded from the internet. It usually consists of six different values for each day of trading. These are:

1. Open - The price of the stock when the market opened on a particular day;
2. High - The highest price the stock reached on a particular day;
3. Low - The lowest price the stock reached on a particular day;
4. Close - The price of the stock when the market closed on a particular day;
5. Volume - The number of shares bought and sold on a particular day;
6. Adjusted Close - The price the stock closed at on a particular day adjusted for dividends and stock splits. This allows the true historical value of shares to be compared over time.

The python module pandas\_datareader was used to retrieve all of the stock data from yahoo finance. The data is returned in a pandas dataframe. An example of the dataframe can be seen in Table 1 and plotted over different time periods in Figure 1 and Figure 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Date** | **Open** | **High** | **Low** | **Close** | **Volume** | **Adj Close** |
| 21/03/2006 | 350.01 | 351.66 | 339.08 | 339.92 | 19735800 | 169.79 |
| 22/03/2006 | 339.75 | 344.1 | 337.5 | 340.22 | 15248800 | 169.94 |
| 23/03/2006 | 342.35 | 345.75 | 340.2 | 341.89 | 14925000 | 170.77 |
| 24/03/2006 | 368.62 | 370.09 | 362.51 | 365.8 | 30474900 | 182.72 |
| 27/03/2006 | 367.09 | 371.71 | 365 | 369.69 | 14100000 | 184.66 |

Table : Five days of trading data for Google (GOOG).

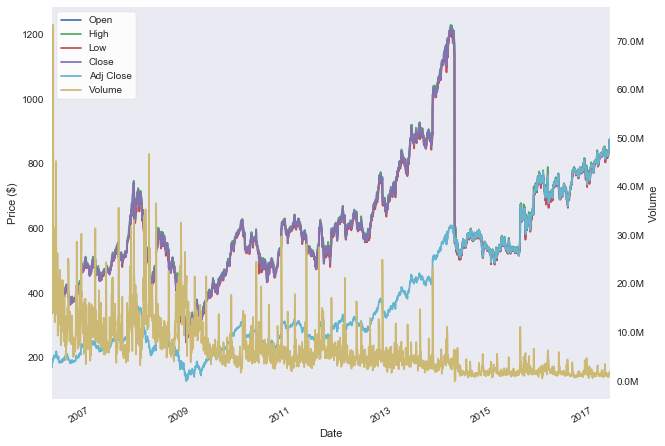


Figure : Google (GOOG) stock data since the 21st March 2006.

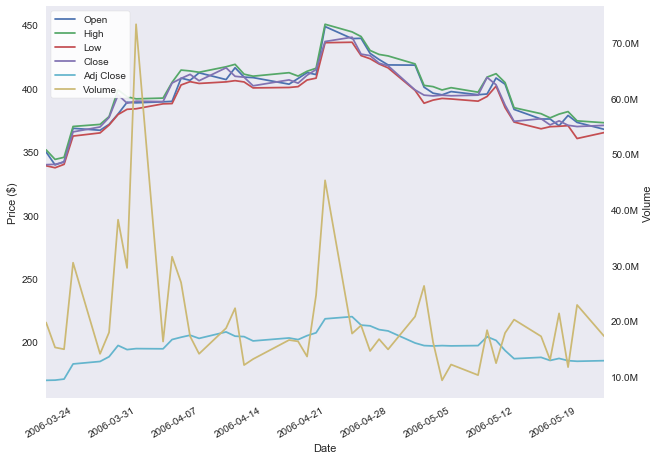


Figure : Google (GOOG) stock data for a two month period.

As seen in Table 1 data is not available for every single day. Data is only available for days that that the stock traded, this will exclude, amongst others, weekends and holidays. Two different strategies were used to deal with this missing data:

1. For days where no stocks trade (i.e. days when the exchange was shut) they can be ignored as only trading days are of interest to our models.
2. For days where only certain stocks didn't trade (i.e. trading was suspended on that stock for some reason) the data will be copied from the previous day. This assumes that the price of that stock did not change on the day it did not trade.

### Twitter Data

Data from twitter consists of posts from users called 'tweets' that are restricted to 140 characters. The python module GetOldTweets has been used to access historical tweets that contain certain search strings. The module has been modified to include the ability to filter by language (in this project only English language tweets have been considered).

GetOldTweets returns a list of tweets (the number is specified in the query). Each tweet contains a number of different parameters such as:

* ID number
* Permalink to the tweet
* Authors username
* Text of the tweet
* Date the tweet was published
* Number of 'retweets'
* Number of favourites
* Users the tweet 'mentions'
* 'Hashtags' in the tweet
* Geolocation of where the tweet was posted

Not every tweet contains every field but only the text of the tweet is of interest for the purpose of this report and that always exists as a tweet isn't a tweet without text.

The text itself is not useful for a machine learning algorithm so the sentiment of each tweets text is calculated using the python module TextBlob . This module calculates whether the text passed to it is of positive or negative sentiment and gives a value in the range -1 to 1 to describe the level of polarity of the text.

## Exploratory Visualisation

### Stock Data

The daily return of a stock is the percentage change in its price from day to day and is defined as:

The classifiers trained during this project will be used to predict if the stock's price goes up or down (i.e. the sign of the daily return). Figure 3 shows an example of the daily return over a long period of time. The plot shows that the daily return is seemingly randomly distributed, it has the following statistical properties:

* mean: 0.00077
* standard deviation: 0.019

Therefore on average the price of the stock raises by 0.077% per day but it does this with a large variance in daily returns.

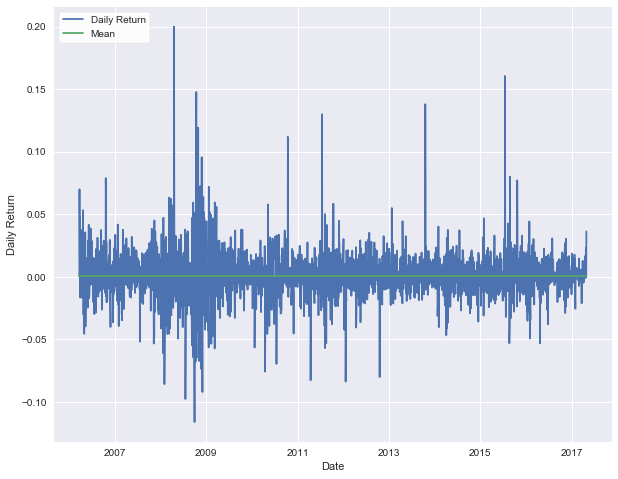


Figure : Google's (GOOG) Daily Return since 21st March 2006.

The classifiers will be used to inform a trading strategy and the ROI will be used to compare different classifiers and strategies. Figure 4 shows the ROI's over time of a number of stocks if a 'buy and hold' strategy was used with the buy taking place on 21st March 2006.

It can be seen that over this time period all of the stocks shown have increased in value but they all had periods where if the stocks had been sold money would have been lost.



Figure : The ROI of various stocks since 21st March 2006.

### Twitter Data

Need to plot average polarity over time or something

## Algorithms and Techniques

use a classifier to determine up or down

requires feature engineering to provide useful inputs

## Benchmark

Buy and hold SPY (tracks the market)

# Methodology

## Data Pre-processing

how to deal with missing data

feature engineering

## Implementation

stocks class

market class

learner class

twitter class

which algorithm

parameters

## Refinement

cross validation

grid search

possible PCA or similar

# Results

## Model Evaluation and Validation

compare basic model before feature engineering, after feature engineering, after grid search

compare to buy and hold spy

## Justification

increased value

compare to SPY

fees generally make it worse

# Conclusions

## Free-Form Visualisation

graph of my models and how they compare to SPY

## Reflection

discuss development process

interesting aspects

difficult aspects

what's the final model like

## Improvement

more features?